

---

# Οργάνωση Υπολογιστών

Single Cycle Processor

Συνοδευτική Αναφορά - Φάσεις #1 #2 #3

---

Πέτρου Δημήτριος - 2018030070

Χανιά, Απρίλιος 2022

Υπεύθυνος εργασίας: Κυπριανός Παπαδημητρίου

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

## 1. Εισαγωγή

Οι 3 πρώτες φάσεις της εργασίας του μαθήματος έχουν σκοπό την πλήρη υλοποίηση ενός επεξεργαστή single cycle, αρχιτεκτονικής παρόμοιας με το MIPS, σε γλώσσα περιγραφής υλικού VHDL. Η σχεδίαση ακολούθησε bottom-up προσέγγιση. Έγινε σταδιακή υλοποίηση των απλούστερων modules και ακολούθησε η σύνδεση τους σε ανωτέρου επιπέδου "κομμάτια υλικού". Για κάθε module που δημιουργήθηκε έχει προηγηθεί το κατάλληλο simulation για να επιβεβαιωθεί η ορθή λειτουργία τους και τυχόν bugs να εντοπίζονται νωρίς ώστε να μην "κληροδοτούνται" σε ανώτερα modules.

## 2. Πλαίσιο υλοποίησης - Specifications

Ο επεξεργαστής υλοποιήθηκε εξ'ολοκλήρου σε VHDL ενώ η ανάπτυξη έγινε στο Vivado 2021.2 της Xilinx. Ο επεξεργαστής διαθέτει 32 καταχωρητές(RF), μονάδα αριθμητικών-λογικών πράξεων (ALU), βαθμίδα ανάκλησης εντολών (IFSTAGE), βαθμίδα αποκωδικοποίησης εντολών (DECSTAGE), βαθμίδα εκτέλεσης εντολών (EXSTAGE) και βαθμίδα πρόσβασης μνήμης (MEMSTAGE). Κάθε μία από τις παραπάνω δομές αναλύεται και παρουσιάζεται στο αντίστοιχο τμήμα της αναφοράς παρακάτω.

## 3. Μονάδες Λογικής

Για την δημιουργία των μονάδων και των βαθμίδων του επεξεργαστή ήταν απαραίτητη η ύπαρξη πολυπλεκτών ή αποκωδικοποιητών, αθροιστών διαφόρων ειδών. Όλες αυτές οι δομές δεν είναι διαθέσιμες άμεσα στη VHDL οπότε ήταν απαραίτητη η υλοποίησή τους. Παρακάτω περιγράφεται η γενική ιδέα δημιουργίας τέτοιων modules σε VHDL:

- Πολυπλέκτης 2 σε 1

Η έξοδος του πολυπλέκτη είναι μια από τις 2 εισόδους του ανάλογα το bit ελέγχου. Με απλότητα αυτό μεταφράζεται σε VHDL σε μια γραμμή εντός του Behavioral:

```
Output <= A when control='0' else B ;
```

- Αποκωδικοποιητής 5 σε 32

Ο αποκωδικοποιητής δέχεται είσοδο πλάτους 5 bit η οποία αντιστοιχεί σε έναν unsigned decimal αριθμό από τον οποίο εξαρτάται ποιο κατά σειρά bit της εξόδου (32b) θα είναι '1'. Σημειώνεται πως μόνο έναν bit της εξόδου μπορεί να είναι '1' για κάθε είσοδο. Η λειτουργικότητα αυτή επιτυγχάνεται κάνοντας το bit που υποδεικνύεται από τον πενταψήφιο αριθμό '1' και τα υπόλοιπα '0':

```
output <= (TO_INTEGER(unsigned(input)) => '1', others=>'0') after 10ns;
```

## 4. ALU

Η μονάδα αριθμητικών-λογικών πράξεων είναι υπεύθυνη για την εκτέλεση των διαφόρων πράξεων μεταξύ τελεστών ή μεταξύ τελεστών και καταχωρητών. Εκτελεί πράξεις όπως η πρόσθεση, η αφαίρεση, η αποθήκευση σε καταχωρητή, OR, AND κτλ. Οι τελεστές αριθμοί είναι σε μορφή 2's complement, αυτό σημαίνει ότι μπορεί να αναπαρασταθεί ένα εύρος αριθμών για n bits:

$$\begin{array}{c} -2^{n-1} \qquad \qquad \qquad 2^{n-1}-1 \\ \hline \qquad \qquad \qquad | \\ \qquad \qquad \qquad 0 \end{array}$$

Εφόσον λοιπόν υπάρχει περιορισμός στο εύρος της αναπαράστασης η ALU διαθέτει έξοδο η οποία αποτελεί ένδειξη υπερχείλισης που προέκυψε κατά την εκτέλεση μιας αριθμητικής πράξης. Η ανίχνευση της υπερχείλισης γίνεται ως εξής:

- Η πρόσθεση προκαλεί υπερχείλιση  
Δύο θετικοί αριθμοί παράγουν αρνητικό αποτέλεσμα ή δύο αρνητικοί παράγουν θετικό αποτέλεσμα, γεγονός που σημαίνει δύο όμοιοι operands παράγουν ετερόσημο result.
- Η αφαίρεση προκαλεί υπερχείλιση  
Δύο ετερόσημοι αριθμοί παράγουν κρατούμενο ετερόσημο του πρώτου τελεστέου, δηλαδή η αφαίρεση μετατρέπεται σε πρόσθεση

Σημειώνεται πως το module της ALU έχει υλοποιηθεί σε procedural κώδικα με `case-when`, πράγμα που δεν είναι ιδανικό. Για τις επόμενες φάσεις θα αντικατασταθεί με Behavioral κώδικα με `when-else`.

## 5. Register File

Το αρχείο καταχωρητών περιέχει τους 32 καταχωρητές του επεξεργαστή και διαχειρίζεται την ανάγνωση και την εγγραφή σε αυτούς. Οι καταχωρητές δημιουργούνται με τη χρήση της `for-generate`. Ο καταχωρητής διαθέτει δυνατότητα ταυτόχρονης ανάγνωσης 2 καταχωρητών και εγγραφής ενός. Στο MIPS ο καταχωρητής R0 έχει σταθερή τιμή 0, προκειμένου να επιτευχθεί αυτό στη εν λόγω σχεδίαση το Write Enable του καταχωρητή R0 συνδέθηκε σε '0'.

## 6. IFSTAGE - Μονάδα Ανάκλησης Εντολών

Η μονάδα Instruction Fetch είναι υπεύθυνη για την ανάκληση των εντολών από την κύρια μνήμη και την τήρηση του Program Counter. Η τιμή του PC αντιστοιχεί στην διεύθυνση της επόμενης προς εκτέλεση εντολής που πρέπει να ανακληθεί από την μνήμη. Το module διαθέτει σήματα όπως Load Enable, ρολόι και reset για τον καταχωρητή PC και σήμα Immed για την εξυπηρέτηση των branches. Ο αθροιστής που φαίνεται στο block diagram δίνει 2 εξόδους, μια έξοδο της μέσης περίπτωσης μεταβολής του PC κατά 4 και μια έξοδο για την περίπτωση που ο PC μεταβάλλεται κατά Immed θέσεις.

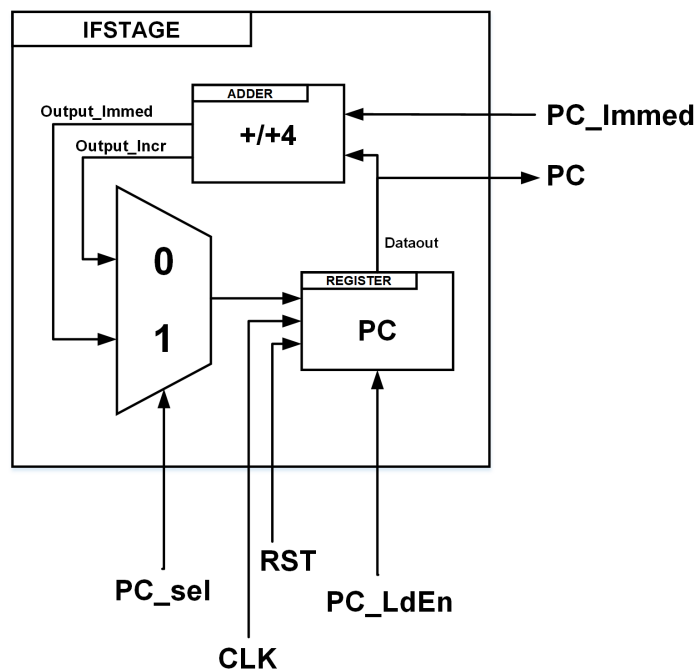


Figure 1: Block diagram του IFSTAGE

## 7. DECSTAGE - Μονάδα Αποκωδικοποίησης Εντολών

Στο στάδιο της αποκωδικοποίησης παράγονται σήματα ελέγχου του datapath και του control σύμφωνα με τις εντολές. Από την επεξεργασία της εκάστοτε εντολής προκύπτουν μεταξύ άλλων:

- Σήματα ελέγχου που έμμεσα θα οδηγηθούν στη μνήμη
- Έλεγχος του αρχείου καταχωρητών και των περιοχομένων του κάθε καταχωρητή
- Διαχείριση και ευθυγράμμιση του immediate
- Σήματα που αντιστοιχούν στους τελεστέους καταχωρητές

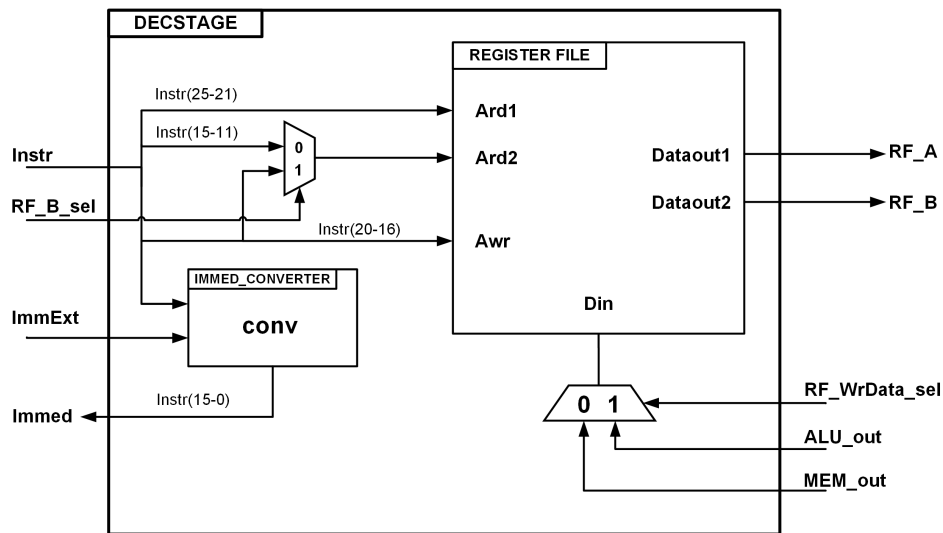


Figure 2: Block diagram του DECSTAGE

Για την κατάλληλη επεξεργασία του immediate δημιουργήθηκε ο IMMED.CONVERTER, ένα module που διαχειρίζεται την ευθυγράμμιση του Immed σε πλάτος 32bit. Διακρίνονται 4 περιπτώσεις επεξεργασίας του Immediate οι οποίες προσδιορίζονται από την τιμή του σήματος ImmExt:

### 1. Zero Fill

Για να προκύψει η 32 bit πλάτους έξοδος το Immediate συμπληρώνεται από 16 μηδενικά bits.

### 2. Sign Extend

Για να προκύψει η 32 bit πλάτους έξοδος το Immediate συμπληρώνεται από 16 bits όμοια με το πρόσημο του, δηλαδή το 1ο bit.

### 3. Sign Extend και Ολίσθηση Αριστερά κατά 2 bits

Χρησιμοποιείται για την μετατροπή των διευθύνσεων μνήμης από 32 σε 11 bits. Πρακτικά είναι πολλαπλασιασμός επί 2.

### 4. Ολίσθηση Αριστερά κατά 16 bits και Zero Fill

Τα 16 bits του Immediate αντιγράφονται στα 16 MSBs της εξόδου και τα υπόλοιπα συμπληρώνονται με '0'.

## 8. EXSTAGE - Μονάδα Εκτέλεσης Εντολών

Το module που υλοποιεί την μονάδα εκτέλεσης εντολών είναι υπεύθυνο για την κατεύθυνση των σημάτων των τελεστών καταχωρητών και του Immed στην ALU. Διαθέτει επίσης εξόδους για τα flags της ALU και του τελικού αποτελέσματος της εντολής.

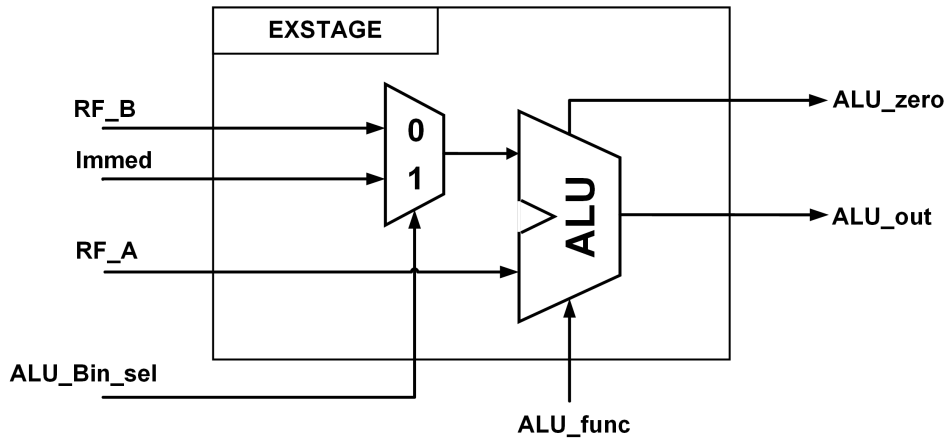


Figure 3: Block diagram του EXSTAGE

## 9. MEMSTAGE - Μονάδα Πρόσβασης Μνήμης

Το MEMSTAGE είναι υπεύθυνο για τη διαχείριση και την επεξεργασία των ζητούμενων διευθύνσεων μνήμης καθώς και για τα δεδομένα εισόδου και εξόδου από και προς αυτήν. Επίσης εντός του MEMSTAGE υλοποιείται η λειτουργικότητα που θα εξυπηρετήσει αργότερα τις εντολές load/store byte και load/store word.

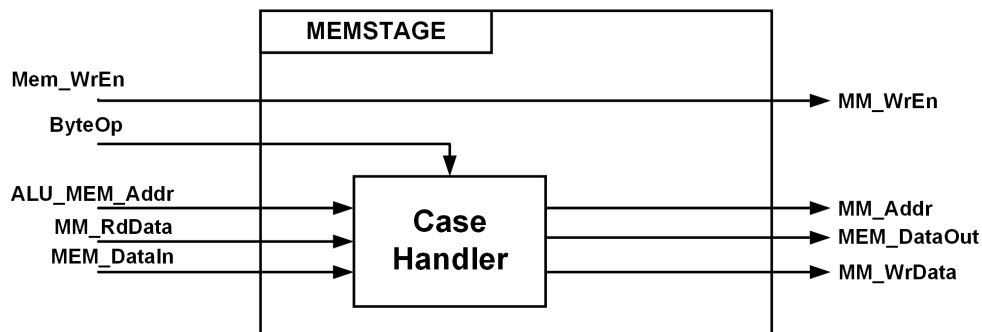


Figure 4: Block diagram του MEMSTAGE

Ο case handler δεν είναι ξεχωριστό module, απλά για λόγους ευταξίας στο διάγραμμα σχεδιάστηκε ως κουτί. Οι λειτουργίες που περιλαμβάνονται στον case handler είναι οι εξής:

- Κατάλληλη πρόσθεση στις διευθύνσεις μνήμης ώστε να ανταποκρίνονται στο data part του stack της μνήμης το οποίο ξεκινάει από την διεύθυνση 0x400 (1024<sub>10</sub>).
- Διαχωρισμός περιπτώσεων load word και load byte
  - **lw**: Τα 32 bits του MM\_RdData μεταφέρονται αυτούσια στο MEM\_DataOut
  - **lb**: Τα 8 LSB bits του MM\_RdData μεταφέρονται αυτούσια στο MEM\_DataOut + ZF
  - **sw**: Τα 32 bits του MEM\_DataIn μεταφέρονται αυτούσια στο MM\_WrData
  - **sb**: Τα 8 LSB bits του MEM\_DataIn μεταφέρονται αυτούσια στο MM\_WrData + ZF

## 10. Datapath

Το Datapath αποτελεί το top-module στο οποίο συνδέονται όλες οι βαθμίδες διαχείρισης των εντολών και της μνήμης. Τα σήματα τα οποία συμμετέχουν σε συναλλαγές μεταξύ των βαθμίδων συνδέονται ενώ τα σήματα που θα συνδεθούν αργότερα στην RAM και στο control module αποτελούν Ports του Datapath.

## 11. Control

Το Control είναι υπεύθυνο για την παραγωγή των κατάλληλων σημάτων ελέγχου ανάλογα την εκτελούμενη εντολή. Μέσω αυτού του module προκύπτουν σήματα για:

- Τον έλεγχο των πολυπλεκτών για κατάλληλη είσοδο ανά τα stages
- Τον έλεγχο των enables για εγγραφή στους καταχωρητές, στη μνήμη και για μέτρηση στον PC
- Την πράξη που θα εκτελεστεί από την ALU
- Τον τρόπο ευθυγράμμισης του Immediate
- Τον έλεγχο για συναλλαγές λέξεων ή byte με τη μνήμη

Με σκοπό να αποφευχθούν μακροσκελή τμήματα κώδικα δημιουργήθηκε ένα bus πλάτους 13 bit εκ των οποίων το καθένα ή ομάδες αυτών αντιστοιχούν στα σήματα ελέγχου. Οι περιπτώσεις διακρίθηκαν με την χρήση when-else statements και σε κάθε μία από αυτές έγινε όσο το δυνατόν καλύτερη απλοποίηση και συγχώνευση των λογικών προτάσεων. Στον κώδικα υπάρχουν επαρκή σχόλια που περιγράφουν την ανάθεση τιμής στο κάθε σήμα ελέγχου.

## 12. RAM

Η μνήμη RAM που συνδέεται με το control module και το datapath module αποτελείται από 2048 θέσεις και χωρίζεται σε δύο μέρη. Στο **TEXT segment**, το οποίο περιλαμβάνει τις εντολές και στο **DATA segment** που περιέχει τα δεδομένα. Επιλέχθηκε το DATA segment να ξεκινάει από τη διεύθυνση  $0x400 = 1024_{10}$ . Στην μνήμη φορτώνεται το **.data** αρχείο το οποίο περιέχει το εκάστοτε πρόγραμμα.

## 13. Επεξεργαστής Ενός Κύκλου - Κυματομορφές

Στο top module **proc\_sc** συνδέονται μεταξύ τους το datapath, το control και η μνήμη RAM. Ουσιαστικό το module αυτό είναι ο επεξεργαστής ενός κύκλου πλήρως υλοποιημένος. Διατίθενται δύο είσοδοι, το ρολόι του επεξεργαστή CLK και το σήμα RESET αφού στην ουσία οι εντολές του προγράμματος φορτώνονται από το **.data** αρχείο και γίνεται η κατάλληλη επεξεργασία τους για την παραγωγή των σημάτων που θα είναι υπεύθυνα για την εκτέλεση τους. Προκειμένου να ελεγχθεί η ορθή λειτουργία της τελικής σχεδίασης δημιουργήθηκαν 3 προγράμματα αναφοράς, εκ των οποίων το καθένα επικεντρώνεται στον έλεγχο συγκεκριμένων εντολών.

- p1: Έλεγχος βασικών immediate και register εντολών
- p2: Έλεγχος branches
- p3: Έλεγχος εντολών συναλλαγών με την μνήμη

Στον φάκελο **WAVEFORMS** υπάρχουν τα τρία προγράμματα αναφοράς και ο assembly κώδικας τους καθώς και screenshots από την προσομοίωση εκτέλεσής τους.

## I. Παρατηρήσεις

- Η υλοποίηση αυτή δεν επιτρέπει το append bytes στην μνήμη με τη χρήση της sb. Όταν γίνεται χρήση της store byte η διεύθυνση πανωγράφεται εντελώς και γράφονται τα καινούρια bytes.
- Περισσότερο αντιπροσωπευτικές για την λειτουργία των datapath και control είναι οι δοκιμές επί των προγραμμάτων αναφοράς χωρίς ωστόσο αυτό να υποβαθμίζει τα single testbenches που έχουν δημιουργηθεί.

## II. Διάγραμμα Single Cycle Processor

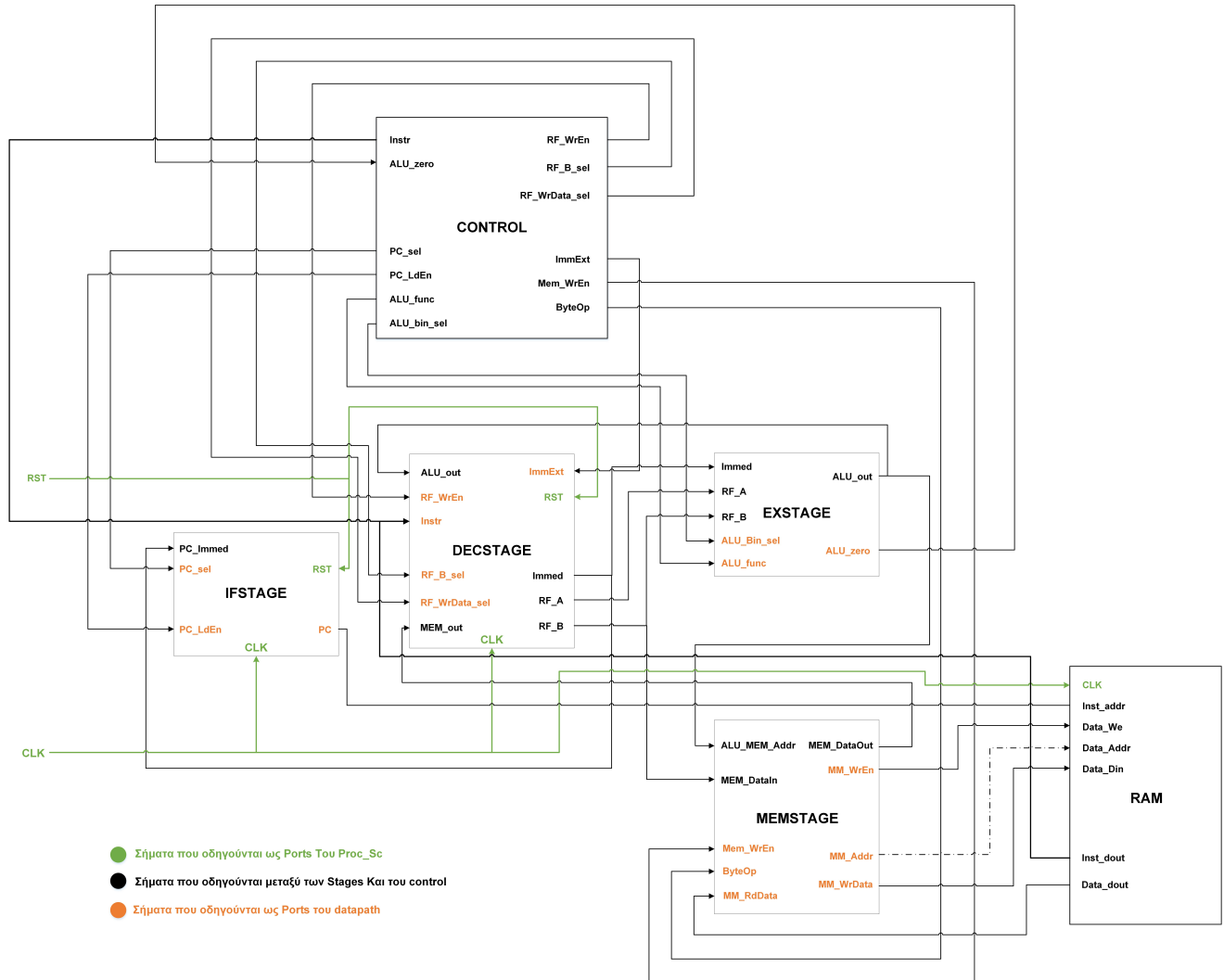


Figure 5: Block diagram του Single Cycle Processor